1-Ч

УДК 004.032.26

Е.И. Чумаченко, В.С. Горбатюк
Национальный технический университет Украины «КПИ», г. Киев
Украина, 03056, г. Киев, ул. Политехническая, 41, lobach21@mail.ru

Алгоритм решения задачи прогнозирования

E.I. Chumachenko, V.S. Horbatiuk
National Technical University of Ukraine "Kyiv Polytechnic Institute"
Ukraine, 03056, c. Kyiv, Polytechnic st., 41

Algorithm of Solving the Forecasting Problem

О.І. Чумаченко, В.С. Горбатюк
Національний технічний університет України «КПІ»,м. Київ
Україна, 03056, м. Київ, вул. Політехнічна, 41

Алгоритм розв'язання задачі прогнозування

В статье рассмотрены основные методы прогнозирования и предложен новый алгоритм, основанный на методе группового учёта аргументов и искусственных нейронных сетях. Алгоритм проверен на реальных данных и показал результаты лучше, чем нейронные сети. Также была найдена экспериментальная зависимость среднеквадратической ошибки прогноза, полученного с помощью данного алгоритма и нейронных сетей, от количества точек в обучающей выборке.
Ключевые слова: прогнозирование, нейронные сети, МГУА, сглаживание.

In the article, the main forecasting methods are described and a new algorithm based on the group method of data handling and artificial neural networks is offered. The algorithm has been tested on the real data and has shown the better results, as compared with neural networks. The experimental dependence of a mean square prediction error, obtained with the use of the offered algorithm and neural networks, on the number of points in the training set is also found.
Key Words: forecasting, neural networks, GMDH, smoothing.

У статті були розглянуті основні методи прогнозування і запропонований новий алгоритм, що базується на методі групового урахування аргументів та штучних нейронних мережах. Алгоритм перевірений на реальних даних і показав результати кращі, ніж нейронні мережі. Також була знайдена експериментальна залежність середньоквадратичної помилки прогнозу, отриманого за допомогою даного алгоритму і нейронних мереж, від кількості точок в навчальній вибірці.
Ключові слова: прогнозування, нейронні мережі, МГУА, згладжування.

Введение

Прогнозирование всегда было и будет одной из наиболее интересующих человечество тем, поскольку знание будущего является, пожалуй, одним из самых заветных его желаний. Однако люди, использующие результаты прогноза, должны понимать опасность выбора неадекватных методов прогнозирования, так как некорректные прогнозы могут привести к принятию неверных решений.

Немалая часть аппарата прогнозирования, используемого сегодня, была разработана в XIX ст. Примером могут служить процедуры регрессионного анализа. Однако с появлением и развитием вычислительной техники начал развиваться и усложняться аппарат прогнозирования – стали появляться всё более сложные методы, требующие всё больших вычислительных мощностей – метод группового учёта аргументов, искусственные нейронные сети.

Целью данной статьи является разработка нового метода прогнозирования, учитывающего и использующего возросшие мощности современной вычислительной техники.

Математическая постановка задачи прогнозирования

Пусть заданы n дискретных отсчетов $\{x_1, x_2, ..., x_n\}$ в последовательные моменты времени $t_1, t_2, ..., t_n$. Тогда задача прогнозирования (рис. 1) состоит в предсказании значения x_{n+k} в некоторый будущий момент времени t_{n+k}, где k – длительность прогноза:

$$x_{n+k} = F(x_1, x_2, ..., x_n),$$

где F – функциональный преобразователь.

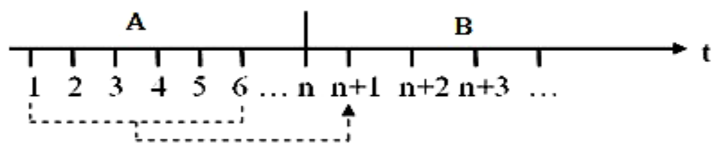


Рисунок 1 – Графическая иллюстрация постановки задачи прогнозирования: A – известные значения; B – прогнозируемый период

Обзор существующих методов прогнозирования

Перечислим основные методы прогнозирования.

Метод скользящего среднего [1]. Метод основывается на простой модели, которая предполагает, что текущее значение y_t, $(t = 1...n)$ ряда $\{y_1, y_2, ..., y_n\}$ является суммой среднего арифметического некоторого количества предыдущих значений и некоторой случайной составляющей.

Метод взвешенного скользящего среднего [1]. Следующим шагом в модификации модели является предположение о том, что более поздние значения ряда более адекватно отражают ситуацию. Тогда каждому значению присваивается вес тем больший, чем более «свежее» значение добавляется.

Метод группового учета аргументов (МГУА) [2]. МГУА – это набор алгоритмов прогнозирования (а точнее, математического моделирования), который основывается на разбиении исходных данных на две выборки: обучающую и проверочную и использовании опорных функций некоторого вида, параметры которых находятся из обучающей выборки, а проверка того, насколько хорошо они моделируют заданный ряд, выполняется на проверочной выборке.

Искусственные нейронные сети (ИНС) [3]. Основной элемент нейронной сети – формальный (искусственный) нейрон. Он представляет собой математическую модель биологической нервной клетки.

ИНС представляют собой систему соединённых и взаимодействующих между собой искусственных нейронов.

ИНС не программируются в привычном смысле этого слова, они обучаются. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными и выходными данными, а также выполнять обобщение.

Способности нейронной сети к прогнозированию напрямую следуют из ее способности к обобщению и выделению скрытых зависимостей между входными и выходными данными. После обучения сеть способна предсказать будущее значение некой последовательности на основе нескольких предыдущих значений и/или каких-то существующих в настоящий момент факторов.

Главным преимуществом ИНС перед другими методами прогнозирования является то, что сети с одинаковым успехом могут прогнозировать процессы, регулярная составляющая которых имеет любой закон изменения значений, в то время как большинство остальных методов лучше всего подходит для процессов, регулярная составляющая которых принадлежит к определенному классу (очевидно, что метод полиномиального сглаживания лучше всего подходит для процессов с полиномиальной регулярной составляющей, метод сглаживания рядами Фурье – для процессов с периодической регулярной составляющей и т.д.). Еще одним важным преимуществом нейронных сетей является возможность обучения.

В данной статье предложен алгоритм прогнозирования, основанный на двух распространенных методах – ИНС и МГУА.

Алгоритм решения задачи прогнозирования

Предложенный алгоритм совмещает в себе преимущества как ИНС, так и МГУА: постепенное увеличение сложности модели МГУА и возможность обучения ИНС, и состоит из следующих шагов:

1. Предварительная обработка исходных данных, включая избавление от выбросов, нормализацию данных (приведение к некоторым границам, например $[-1...+1]$) и т.д. [4-7]. Этот этап часто является важнее этапа моделирования. Также следует учитывать специфические характеристики прогнозируемого процесса, как-то сезонность значений в случае предсказании различных атмосферных показателей, трендовую составляющую в большинстве финансовых процессов и т.д.

В работе [6] показана важность избавления от ненужных выбросов в исходных данных при использовании искусственных нейронных сетей. Очень важно определить, является ли данное значение случайным, и, следовательно, ненужным выбросом, или же этот выброс является информативным. Для этого следует учитывать сам прогнозируемый процесс, например, если требуется построить модель энергопотребления некоторого региона в зависимости от дня недели, то следует учитывать, что энергопотребление по выходным дням будет резко отличаться от потребления по будним дням, и, следовательно, значения потребления в выходные дни не будут являться ненужными выбросами. В этом случае вполне разумным будет при обучении сети подавать на её входы также бинарное значение, которое будет равно единице в случае выходного дня, и нулю в обратном случае.

Если же известно, что прогнозируемый процесс не содержит информативных выбросов, то существует несколько основных алгоритмов избавления от случайных выбросов:

I. Простейший алгоритм, основанный на характеристиках случайной величины, согласно которому выбросами считаются все значения, отклонённые от среднего на величину, большую, чем 2…3 среднеквадратических отклонений σ^2.

II. Алгоритм Tukey 53H [7], который заключается в построении сглаженной последовательности, используя медианный фильтр и фильтр скользящего среднего, после чего выбросами будут считаться все исходные значения, отклонение которых от значений сглаженной последовательности больше чем некоторый наперед заданный порог k.

2. Из данных вида $X = \{x_n, n = 1 \ldots N\}$, где N – количество отсчётов, полученных в результате предыдущего этапа обработки, методом скользящего окна формируются две следующие матрицы:

$$X^{(0)} = \begin{bmatrix} x_1 & x_2 & \dots & x_k \\ x_{k+2} & x_{k+3} & \dots & x_{2*k+1} \\ \dots & \dots & \dots & \dots \\ x_{N-k-1} & x_{N-k} & \dots & x_{N-1} \end{bmatrix}, y = \begin{bmatrix} x_{k+1} \\ x_{2*k+2} \\ \dots \\ x_N \end{bmatrix},$$

где k – размер окна. С помощью этих матриц в дальнейшем будет осуществляться обучение сетей – каждый вектор-строка $\overline{x}_n = \{x_{n1}, x_{n2}, \dots, x_{nk}\}, n = 1\dots m$ (где m – количество строк матрицы) матрицы $X^{(0)}$ и соответствующее ему значение y_n являются независимым примером, а вектор-столбец $\vec{x}_z = \{x_{z1}, x_{z2}, \dots, x_{zm}\}, z = 1\dots k$ – отдельной переменной.

3. Полученные примеры делятся в некотором соотношении (обычно $0.7m : 0.3m$) на обучающую и проверочную выборки.

4. Определяется вид опорных функций, а именно от каких переменных они будут зависеть, например $f_l = f(x_i, x_j)$ или $f_l = f(x_i, x_j, x_i * x_j, x_i^2, x_j^2)$, где $l = 1\dots C_k^2$ – количество вариантов выбора двух переменных из k возможных (в общем случае $l = 1\dots C_k^o$, где o – количество переменных, отбираемых как аргументы опорных функций). В отличие от МГУА, где также определяется конкретный вид функций (например $f_l = a_0 + a_1 * x_i + a_2 * x_j$) в данном алгоритме выбираются лишь переменные, от которых будут зависеть опорные функции (степени переменных и их ковариации можно представить как дополнительные переменные), а нахождение зависимости значения функции от переменных будет осуществляться нейронной сетью.

5. Составляются C_k^o (o – количество переменных, учитываемых в опорных функциях, для описанного выше примера o = 2) простых многослойных персептронов (МП) с одним выходом, одним скрытым слоем с малым количеством нейронов (около 3) в нём, и нужным количеством входов (для опорных функций вида $f_l = f(x_i, x_j)$ сеть должна иметь 2 входа, для $f_l = f(x_i, x_j, x_i * x_j, x_i^2, x_j^2)$ – 5 входов и т.д.).

6. Каждый персептрон сопоставляется с конкретной опорной функцией, а именно выбираются переменные, которые будут подаваться на входы сети (например, для опорных функций вида $f_l = f(x_i, x_j, x_i * x_j, x_i^2, x_j^2)$ один МП будет работать с переменными $x_1, x_2, x_1 * x_2, x_1^2, x_2^2$, а другой – с переменными $x_1, x_k, x_1 * x_k, x_1^2, x_k^2$), и обучается, используя только примеры из обучающей выборки.

7. На этом шаге следует составить исходные данные для следующей итерации алгоритма. Для этого следует определить среднеквадратическую ошибку (СКО) каждого МП на проверочной выборке и отобрать k лучших (также можно отобрать меньшее количество, но дополнив их исходными переменными, которые были входами сетей с малой СКО) сетей, после чего составить новую матрицу:

$$X^{(1)} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1k} \\ h_{21} & h_{22} & \dots & h_{2k} \\ \dots & \dots & \dots & \dots \\ h_{m1} & h_{m2} & \dots & h_{mk} \end{bmatrix},$$

где h_{ij} – значение выхода j-й из k лучших сетей при подаче на её входы i-го примера, $i = 1\dots m, j = 1\dots k$ (либо исходная переменная).

8. Выполняется следующая итерация, но в качестве исходных примеров берется уже матрица $X^{(1)}$. Итерации выполняются, пока значение СКО сетей на проверочной выборке уменьшается, либо пока не будет достигнута требуемая СКО.

9. В ходе алгоритма на каждой итерации следует запоминать структуру и веса сетей (и/или исходных переменных), которые были отобраны для получения матрицы исходных примеров для следующей итерации. После достижения требуемой СКО (либо достижения итерации, после которой СКО начинает расти) выполнение итераций прекращается, и отбирается единственная сеть, которая в дальнейшем и будет прогнозировать окончательное значение. Для дальнейшего прогнозирования некоторого значения x_c с использованием полученных результатов требуется:

I. Составить исходный пример $inp = \{x_{c-k-1}, x_{c-k}, \ldots, x_{c-1}\}$.

II. Подать данный пример на входы тех МП, которые были использованы для получения матрицы исходных примеров второй итерации, и, используя полученные значения выходов (и/или исходных переменных), составить новый входной пример для второй итерации.

III. Повторять второй шаг пока не будет составлен входной пример для итерации, на которой была совершена остановка алгоритма, после чего подать составленный пример на входы отобранного МП, выход которого и будет прогнозом значения x_c.

Преимущества и недостатки алгоритма

Преимуществом данного алгоритма, по сравнению с МГУА, является то, что не требуется явно задавать вид опорных функций, требуемую зависимость будут находить нейронные сети, которые, как известно, очень хорошо справляются с этой задачей.

Данный алгоритм также лишен некоторых недостатков ИНС: во-первых, при построении сети заранее неизвестна её оптимальная сложность, а во-вторых, слишком простая нейронная сеть может не полностью смоделировать прогнозируемый процесс, а слишком сложные сети склонны к так называемому «оверфиттингу», или переобучению, в результате которого сеть начинает моделировать шум, присутствующий в обучающей выборке, и как следствие показывает плохие результаты на проверочной выборке. Предлагаемый же алгоритм на каждой итерации использует простые сети, не склонные к переобучению, но за счет каскадного усложнения способен прогнозировать очень сложные процессы.

Главным недостатком предложенного метода является его ресурсоемкость, так как на каждой итерации строится некоторое количество нейронных сетей, что может потребовать немало времени. Но за счет использования численно-оптимизированных алгоритмов обучения нейронных сетей и общей оптимизации алгоритма, можно достигнуть сравнительно быстрой работы метода.

Результаты применения изложенного алгоритма

Для проверки предложенного алгоритма был использован публично доступный набор данных годовых цен на медь с 1800 по 1997 год [8].

Для конкретной реализации алгоритма использовались следующие параметры:

1. Размер скользящего окна для получения матриц исходных примеров $k = 5$.
2. Соотношение размеров обучающей и проверочной выборки – $0.7m : 0.3m$, где m – общее количество исходных примеров, причем для составления обучающей выборки использовались первые $0.7m$ примеров;
3. Вид опорных функций – $f_l = f\left(x_i, x_j, x_i * x_j, x_i^2, x_j^2\right)$, $l = 1 \ldots C_5^2$, $i = 1 \ldots 5, j = 1 \ldots 5, i \neq j$;

4. Структура МП: один выход, один скрытый слой с 3 нейронами, 5 входов;
5. СКО для отбора сетей вычислялась на всех исходных примерах, то есть на проверочной и обучающей выборке;
6. Для создания матрицы исходных примеров для следующей итерации брались выходы 3 сетей с минимальной СКО на обучающей и проверочной выборке, и 2 исходных переменных, которые являлись входами сети, показавшей наименьшую СКО.

Для сравнения результатов работы алгоритма также был построен многослойный персептрон с одним выходом, 2 скрытыми слоями, с 8 нейронами в первом слое и 5 во втором, и 5 входами. МП, как и предложенный алгоритм, обучался лишь на обучающей выборке. Поскольку известно, что ИНС также склонны к «застреванию» в локальных минимумах функции ошибки сети, было обучено около десятка МП, и выбран персептрон с минимальной СКО.

Сравнение результатов, полученных при использовании ИНС и данного алгоритма без предварительной обработки данных и с предварительной обработкой, показано на рис. 2 и рис. 3 соответственно.

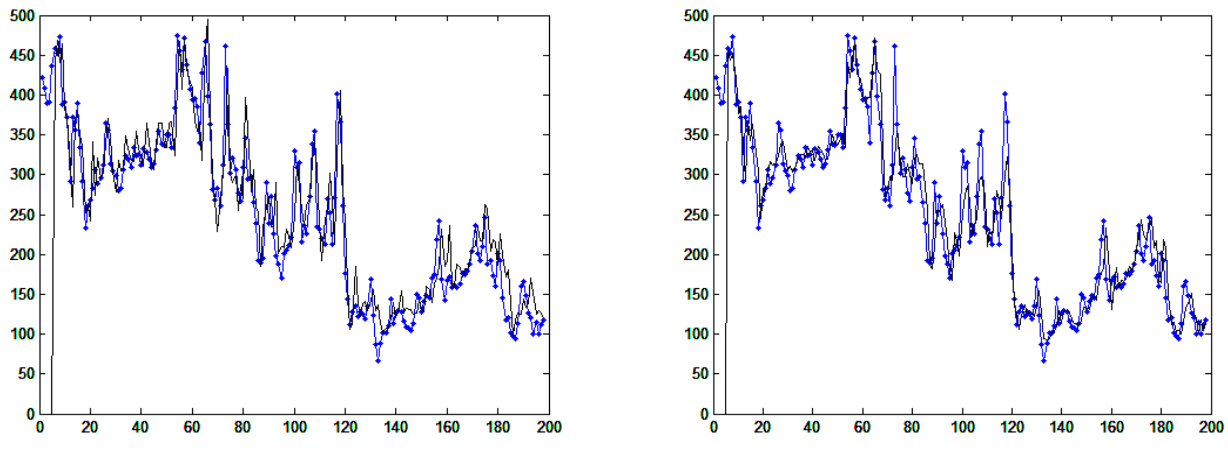


Рисунок 2 – Сравнение результатов прогноза ИНС (слева) и предложенного алгоритма (справа) (без предварительной обработки данных).
– прогноз, – исходные значения

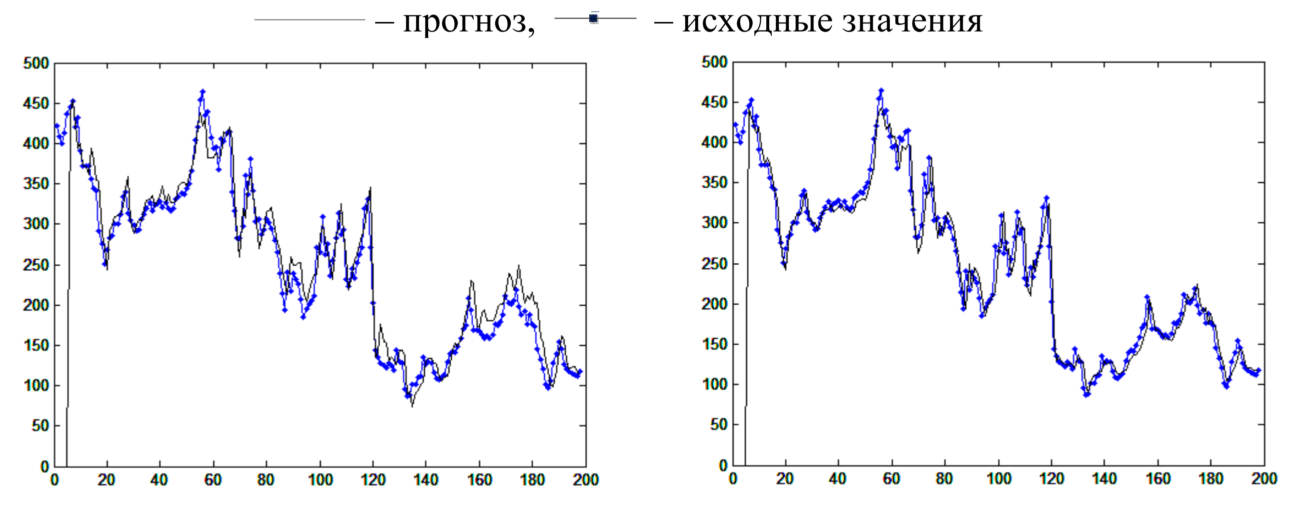


Рисунок 3 – Сравнение результатов прогноза ИНС (слева) и предложенного алгоритма (справа) (с предварительной обработкой данных).
– прогноз, – исходные значения

В качестве алгоритма предварительной обработки данных использовался ранее описанный алгоритм Tukey 53H. Значение СКО без предварительной обработки данных: ИНС – 1256.5, предложенный алгоритм – 920.0306; с предварительной обработкой: ИНС – 507.0927, алгоритм – 355.4813.

Из обоих рисунков видно, что обученный персептрон хуже прогнозирует значения проверочной выборки (самые поздние значения), в то время как предложенный алгоритм одинаково хорошо прогнозирует значения как обучающей, так и проверочной выборки, хотя для построения сетей использовалась лишь обучающая выборка, что говорит о более точной модели процесса.

К тому же алгоритм показал значительно лучшую СКО при прогнозировании без предварительной обработки данных, что говорит о возможности его применения для прогнозирования зашумленных данных.

Также, для сравнения работы предложенного алгоритма и ИНС при различном количестве точек в обучающей выборке были вычислены СКО алгоритма и ИНС на всей исходной выборке, полученные после их обучения на обучающей выборке с изменяющимся количеством примеров в ней. Результаты приведены в табл. 1 и представлены графически на рис. 4:

Таблица 1 – СКО алгоритма и ИНС при различном количестве точек в обучающей выборке

K	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
Алгоритм	2430	1749	3199,5	2093,8	1275,4	1239,7	1106	994,901	950,03
ИНС	5801,7	10230	7688,6	3445,2	1531,2	1903,3	1237,4	1226,9	1024,5

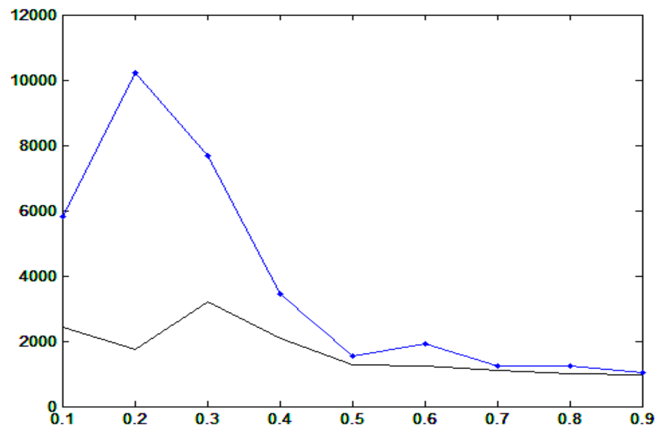


Рисунок 4 – Графическое представление СКО алгоритма и ИНС в зависимости от количества точек в обучающей выборке. ——— – СКО алгоритма,

—•— – среднеквадратическая ошибка ИНС

В обучающую выборку отбирались первые K*m примеров, где $K = 0.1 \ldots 0.9$, m – общее количество исходных примеров (193 для данного случая). Как видно на рис. 1, алгоритм даёт значительно меньшую СКО при малом количестве точек в обучающей выборке, что также говорит о более адекватной модели процесса.

Также можно заметить, что начиная с некоторого значения K СКО как сети, так и алгоритма резко возрастают, что свидетельствует о недостаточной информативности обучающей выборки, и как следствие неполной модели.

Выводы

Предложенный алгоритм показал лучшие результаты по сравнению с искусственными нейронными сетями, как в условиях зашумленности данных, так и при малом количестве обучающих примеров, что говорит о целесообразности его дальнейшего использования для задачи прогнозирования.

Литература

1. Алесинская Т.В. Методы скользяшего среднего и экспоненциального сглаживания / Т.В. Алесинская // Экономико-математические методы и модели : уч. пособие по решению задач по курсу. – Таганрог : Изд-во ТРТУ, 2002. – 153 с.
2. Ивахненко А.Г. Метод групового урахування аргументів – конкурент методу стохастичної апроксимації / А.Г. Ивахненко // Автоматика. – 1968. – № 3. – С. 58-72.
3. Мак-Каллок У.С. Логическое исчисление идей, относящихся к нервной активности / У.С. Мак-Каллок, В.Питтс // Автоматы ; [под ред. К.Э. Шеннона и Дж. Маккарти]. – М. : Изд-во иностр. лит., 1956. – С. 363-384.
4. Amir F.A. A comparison between neural-network forecasting techniques – case study: river flow forecasting / F.A. Amir, I.S. Samir // IEEE Transactions on neural networks. – 1999, march. – Vol. 10, № 2. – С. 402-409.
5. Mohsen H. Artificial neural network approach for short term load forecasting for Illam region / H. Mohsen, S. Yazdan // World Academy of Science, Engineering and Technology 28 2007. – С. 280-284.
6. Jerome T.C. neural networks and robust time series prediction / T.C. Jerome, R.M. Douglas, L.E. Atlas // IEEE transactions on neural networks. –1994, march. – Vol. 5, № 2. – С. 240-254.
7. Klevecka Irina. Pre-Processing of Input Data of Neural Networks: The Case of Forecasting Telecommunication Network Traffic / Irina Klevecka, Janis // Telektronikk 3/4.2008. – С. 168-178.
8. http://robjhyndman.com/TSDL/micro-economic/.

Literatura

1. Alesinskaja T.V. Metody skol’zjashego srednego i jeksponencial’nogo sglazhivanija. Uch. posobie po resheniju zadach po kursu “Jekonomiko-matematicheskie metody i modeli”. Taganrog: Izd-vo TRTU. 2002. 153 s.
2. Ivahnenko A.G. Avtomatika. 1968. № 3. S. 58-72.
3. Mak-Kallok U. S. Sb.: “Avtomaty” pod red. K. Je. Shennona i Dzh. Makkarti. M.: Izd-vo inostr. lit. 1956. S.363-384.
4. Amir F. A. IEEE Transactions on neural networks. Vol. 10. № 2. March 1999. S. 402-409.
5. Mohsen H. World Academy of Science, Engineering and Technology. 28. 2007. S. 280-284.
6. Jerome T. C. IEEE transactions on neural networks. Vol. 5. № 2. March 1994. S. 240-254.
7. Irina Klevecka. Telektronikk 3/4. 2008. S. 168-178.
8. http://robjhyndman.com/TSDL/micro-economic/.

E.I. Chumachenko, V.S. Horbatiuk

Algorithm of Solving the Forecasting Problem

The purpose of the article is to introduce the new method of solving the forecasting task. As a base of this method, artificial neural networks and group method of data handling were used. The main idea is to use artificial neural networks instead of predefined functions at each iteration of the group method of data handling.

The advantages of this approach is that there is no need to define a type of functions to be used in forecasting and also that only artificial neural networks with a simple structure are used at each iteration thus preventing over fitting. The main disadvantage is that a lot of computational resources are needed to handle creation and training of many neural networks.

The algorithm was tested on real data (open dataset of yearly copper prices, 1800-1997) and showed better results in comparison with neural networks, especially on the testing set, which usually means better generalization (total normalized MSE on the training and on the testing set of the artificial neural networks was 1.257 in comparison to the algorithm’s MSE of 0.920).

The experimental dependence of a mean square prediction error, obtained with the use of offered algorithm and neural networks, on the number of points in the training set is also found. The algorithm has shown even better result in comparison with neural networks when the number of points in a training set became smaller.

Статья поступила в редакцию 27.04.2012.